Ontological Approach for E-Services

Li Li and Yun Yang
Faculty of Information and Communication Technologies
Swinburne University of Technology
PO Box 218, Hawthorn, VIC 3122, Australia
{lli, yyang}@ict.swin.edu.au

Abstract

Information sharing is crucial to the success of business applications in a collaborative environment. This paper presents a novel approach for information sharing which is suitable for distributed computing. Agent technologies are applied to cope with service agents' on and off and to provide semantic support for implementation of complex e-business applications that may span diverse organisations. A finite state machine (FSM), which is ideally suited for modelling business processes, is investigated to facilitate process automation. Our approach aims to support dynamic information sharing and reuse in e-business applications. A case study illustrates our ontological approach in employing FSM agents in an inter-organisational environment. A demonstration integrating the JADE platform and frame-based ontologies is also presented.

Keywords: Ontology, Multi-agent systems, Finite state machine, Process automation

1 Introduction

One of the key characteristics of Internet based e-business world is enterprise applications or integration. With ever-increasing number of business to enter into e-business, there is a tendency to automate the processes for the purpose of increasing competitiveness by taking a customer-centric paradigm. Until now, e-business has been concerned and designed mainly for human processing. However, the next generation of an e-business process platform aims at the machine-processable information that enables diverse enterprises integration. The tasks of such an e-business platform are challenging. Web services and the Semantic web are the two recent developments which enable intelligent services, and offer greater functionality and interoperability than current stand-alone services.

On one hand, Web services are software components that use standard Internet technologies to interact with one another dynamically. In this model, business offers Web services that applications running in other business could invoke automatically, which require extensive integration and development efforts to build bridges between systems. The Semantic web, on the other hand, aims at establishment of a different level of interoperability that not only defines a syntactic form of e-business, but also a semantic content. The potential benefits of these technologies are obvious. Web services provide a platform of e-business, and Semantic web supports the intelligent business transaction mechanism to achieve processes automation. Recent W3C standardisation efforts like RDF/RDFS and OWL facilitate semantic interoperability, and several leading organisation proposals offer XML-based Web service specifications and standards that provide the building blocks for Internet based e-business.

While the technology development of the Semantic Web and the convergence of Web services are long term efforts, it is time to study e-business process automation based on existing technologies to reach a common understanding about e-business from newly developed technologies. This paper is motivated by these technologies for Internet based e-business processes, but does not attempt ambitiously to propose another standard of e-business. Rather, it proposes an innovative e-business process modelling approach with respect to various proposed standards and their building blocks that ontologically represents e-business processes and semantically infers among different business process standards to achieve automation.

In the proposed approach, we start with the goal of understanding e-business processes in existing e-business standard proposals. The semantic meanings and relations of these processes are exploited and constructed as an e-business ontology. Technically, a frame-based model is considered for describing the ontology. Furthermore, agent systems, equipping with a part or full of the domain ontology and inference mechanisms, are considered as a business process platform to achieve e-business process automation, especially for the information sharing purpose. We concen-

1-4244-0963-2/07/$25.00 ©2007 IEEE.

trate on the approach itself in this paper. Please refer to our other papers [22, 23] for more specific information on process modelling.

The rest of the paper is organised as follows. Section 2 discusses technical background of an ontology, agent technologies, and a finite state machine to be used in an e-business environment. Section 3 presents business process supporting protocols. Section 4 illustrates an order fulfillment process and shows how the proposed approach can be used in process automation with information sharing, so to support a wide range of processes in e-commerce. Finally, Section 5 summarises our work and presents future work.

2 Technical Backgrounds for Business Automation

2.1 Ontology for E-business Processes

Ontology is defined as an explicit specification of conceptualisation for the purpose of enabling knowledge sharing and reuse [5]. It is a description (like a formal specification of a program) of the concepts and relationships. The aim of an ontology is to capture certain characteristics of the world by defining meta-concepts and meta-relations and filling each catalogue with terms and relations.

Ontologies for business process are crucial to run business in today's dynamic, complex and heterogeneous e-business environment. Although a few proposals such as SUMO (http://ontology.teknowledge.com/), Enterprise Ontology(http://www.aiai.ed.ac.uk/project/enterprise/), TOVE (http://www.eil.utoronto.ca/enterprise-modelling/tove/), and MIT Process Handbook (http://process.mit.edu/), etc. bring freshness to the state of the art in ontologies. Some of them are defined in highly conceptual abstraction outlining few activities of processes while others focus on detailed elements of business modelling. However, none of them deal with e-business processes from the run-time perspective of what the essential characteristics of Internet e-business are. Therefore, it is necessary to develop relevant techniques that are specifically tailored to e-business to support information sharing and ultimately to achieve process automation.

According to Papazoglou [17], e-business applications are based on the existence of standard ontologies for a vertical domain that establish a common terminology for sharing and resue. An ontology consisting of terminology and corresponding relationship descriptions establishes a common vocabulary for participating agents to share consistent business semantics across different market domains and segments. It can be used as a guideline to describe requirements and business concepts [8]. Ontology is nothing new, it has been a topic in knowledge engineering and artificial intelligencefor decades. Successfully combining ontologies and software agents, especially in Multi-Agent Systems (MAS) to facilitate coordination and cooperation between agents [3, 20], is a spur for advancements in ontology-based applications.

The use of ontologies in this context requires a well-designed, well-defined, and Web-compatible ontology language with supporting reasoning tools. The syntax of this language should be both intuitive to human users and compatible with existing Web standards. A frame-based representation is ideal to describe an ontology with formally defined semantics and sufficient expressive power. A frame-based language RDF/RDFs will be used in this paper.

2.2 Software Agents for E-business Processes

The presence of agent technologies becomes so powerful that it allows involved agents to be more flexible and credible in their ways to model business processes with interaction protocols. A multi-agent system (MAS) is a computational environment that is well suited for analysing coordination problems involving multiple agents with distributed knowledge. A MAS aims at the distributed, heterogeneous and autonomous properties of a system for real world problems. Thus a MAS model seems to be a natural choice for e-business process automation, which is intrinsically dealing with coordination and coherence among multiple agents.

In fact, agent technologies have been reported to be used widely in the real world cases. For example, P&G [18] plans for an agent-enabled supply network in 2008 based on its current advantageous agent-based supply chain network.

From the literature [1, 2, 4, 6, 16, 19], agents have been used to support virtual enterprises (VEs) in complex business applications by taking advantages of agents which act automatically with no intervention from other entities such as humans or computer systems, having control over its own actions to achieve the goals. We have experienced ontologies and agent technologies in VE's formation in [10].

It is evident that deploying agent technology will enable organisations to perform what-if analyses and lead to benefits particularly in cost savings, inventory reduced and better customer service.

In this paper, we intent to model Internet based e-business processes by developing software agents and ontologies to achieve e-business process automation. Agents' behaviours will be governed by finite state machines discussed next.

2.3 Finite State Machines

Most business process designs and implementations are based on the sequential models (*http://orchestrationpatterns.com/*). However, as far as process concurrency is concerned, the sequential model is obviously insufficient. Examples can be found in processes where two or more paths[1] can execute completely independent of other paths. For instance, in an order fulfillment process, billing and shipping are executed independently given different input strings. Moreover, it is impossible, or at least very hard, to cope with the process with multiple complex looping conditions as every step in a path of execution is less determined at design-time. In this sense, it is most likely that the sequential models can hardly be applied in highly customer-centric processes modelling, where an execution path is highly dependent on the input and is likely uncertain at design-time.

A finite state machine (FSM) model is an ideal candidate in this regard. The process in FSM is defined as any processing function. The behaviour of a process acts as an FSM which is consisting of all execution paths. From a broader perspective, an FSM is a model of computation, which is defined as a six tuple $(\Sigma, \Gamma, S, s_0, \delta, \omega)$ consisting of:

- an input alphabet Σ
- an output alphabet Γ
- a set of states S
- an initial state s_0 which is an element of S
- a transition function $\delta : S \times \Sigma \longrightarrow S \times \Gamma$
- a output function ω

The advantageous features of an FSM in the process modelling are summirised as follows:

- It is a very straightforward way to understand the execution of a process.
- Loops and parallel executions are included in an FSM.
- An FSM is very applicable when there are multiple entries/exits from a process, and the behaviour of an entry is dependent on the process state.
- It is feasible to augment FSMs when needed.

It is apparent that an FSM allows the design and implementation of business processes to be more flexible by taking multithreaded computing, transition rules and processing logic into consideration. Since a process has an FSM nature, it is convincing that an FSM is a good model for process modelling.

On one hand, we have experience [13, 14, 15] in developing ontologies and agents, and in applying agents to achieve process automation [23]. On the other hand, the work carried out at the Multi-Agent Systems Laboratory at the University of Massachusetts at Amherst (*http://dis.cs.umass.edu/research/fsm/fsmcc.html*) has set a good example to develop FSM agents. Our work on FSM agents for Internet based e-business is inspired by the achievement in this MAS Laboratory. In saying so, we attempt to explore process automation based on these technologies in order to support a wide range of applications in the real world.

[1] A path in a finite state machine is defined as starting at an initial state, ending at a terminal state, and traversing one transition at a time.

Figure 1. Sequence in FSM

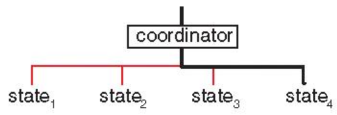

Figure 2. Thread of execution in FSM

3 Finite State Machine Agent for Process Modelling

Generally speaking, a process consists of basic elements such as sequence, concurrence, condition and iteration. In terms of process, these elements can be further described as follows:

- Sequence: the sequentially conducted operations. Figure 1 is an illustration, where states $state_1$, $state_2$, $state_3$ and $state_4$ are executed logically.
- Concurrence: the synchronisation of two or more parallel operations. Figure 2 is an illustration, where states $state_1$, $state_2$, $state_3$ and $state_4$ are executed independently. In an FSM implementation, a coordinator (agent) determines which state to run. For example, the thick line in Figure 2 denotes the thread of execution.
- Condition: choices between two or more execution paths. The coordinator makes a decision based on different cases. The behaviour of an FSM can vary from time to time. For the demonstration purpose, the next run may correspond to a particular path along $state_2$ instead of $state_4$ in Figure 2.
- Iteration: A certain path executed more than once depend upon received strings and the current state. An example in Figure 3 shows that the thick path with $state_4$ runs repeatedly.

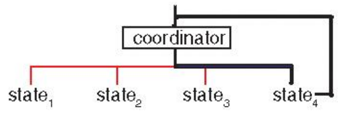

Figure 3. Iteration in FSM

The next state in an FSM is determined by a transition function $\delta : S \times \Sigma \longrightarrow S \times \Gamma$ as discussed in Section 2.3. In terms of transition function of an FSM, predicate calculus can be used to describe FSM agents' behaviours under the ontological view.

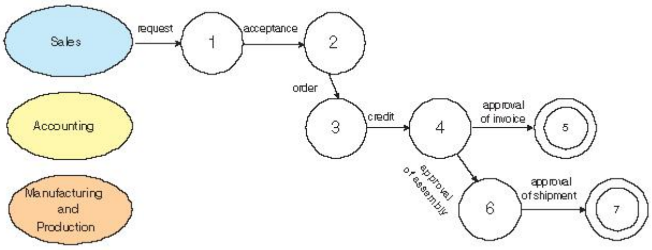

Figure 4. An order fulfillment process in FSM

With the frame-based ontologies, processing logic in an FSM agent can be represented with predicate calculus. The logic implication is described as follows:

Logic implication: $\forall x, y(\exists z)R_i(?x, ?y) \wedge R_j(?y, z) \longrightarrow R_k(?x, z)$ for transitivity, reflectivity and antisymmetry relations.

• Example 1: Concepts and relationships:
Assume relationships R_i, R_j, R_k are "*part-of*", then the formula $\forall x, y(\exists z)R_i(?x, ?y) \wedge R_j(?y, z) \longrightarrow R_k(?x, z)$ exists.

• Example 2: Processing logic:
Assume relationships R_i, R_j, R_k are "*confirmation*", "*approval*", and "*assembly*", respectively, then the formula $\forall x, y(\exists z)R_i(?x, ?y) \wedge R_j(?y, z) \longrightarrow R_k(?x, z)$ exists.

Combining with logic assertion in predicate calculus, a particular implication rule will be "fired" to respond in a timely fashion to evolving e-business processes.

4 Case Study

An order fulfillment process encompasses activities from Sales, Accounting, and Manufacture and Production [9]. It is illustrated in Figure 4.

Few constraints are posed on the agent implementation in order to pursue flexibility. In most cases, agents are not forced to commit to a single ontology. However, it is necessary for participating agents to conform to the most general concepts in an application domain. For example, as far as the order fulfillment process is concerned in this case, some general terms and relations are essential. This general ontology, which consists of mostly used concepts, acts as a baseline to enable agents to participate in this virtual community. Additionally, the agent has domain ontologies which enable it to perform reasonably under certain circumstance. When an agent joins this community, it declares to the community what ontologies it can interpret. At the same time, ontology mapping mechanisms [12] are running at the background to allow registered agents to create ontology mapping at run-time. By doing so, agents with different ontologies can share their perspectives and eventually reach an agreement.

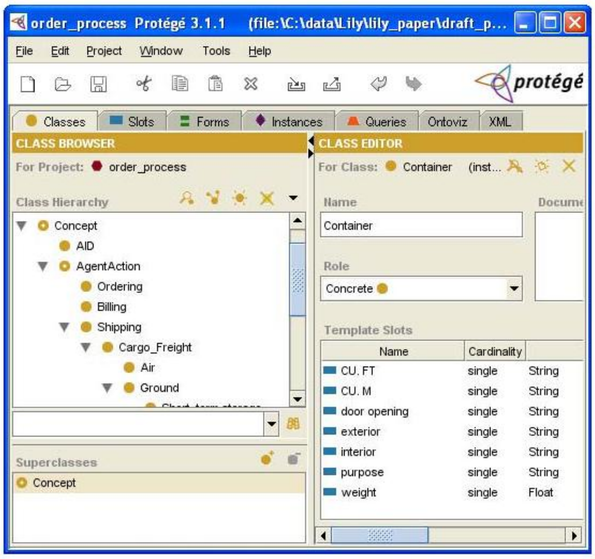

Figure 5. An order fulfillment ontology in Protégé

4.1 General ontology

A general ontology is about the definition of features which are common to all agents involved in communication. Essentially agents commit to a same domain-independent ontology. In the above case, the most shared commonalities are demonstrated in Protégé[2] in Figure 5, respectively.

For example, the "Shipping" includes "Cargo_Freight", "Ocean", and "Ground" by using the equipment "Container", where "Container" is defined as another general concept with particular features.

Usually, only one general ontology is insufficient. Domain-dependent ontologies are required for a specific domain. For example, there is little doubt that there has difference between international and domestic markets in terms of shipment. As for an international shipment, apparently policies, tariffs, customer services, and regional information are different from those in domestic markets. In this regard, agents may take advantages of domain-dependent ontologies as well as general ontology. In this case, an order fulfillment process can be instantiated in a specific domain where domain-depended ontologies will assist agents to acquire more information in addition to general ontologies.

[2]Protégé is an open source ontology editor and knowledge-base framework. For more information, please visit the website at *http://protege.stanford.edu/*.

4.2 FSM

From an FSM perspective, no matter what paths the process transits to, a process maps perfectly into an FSM. The transitions between states are exactly depicted by the state transition function $\delta : S \times \Sigma \longrightarrow S \times \Gamma$. The order fulfillment process is mapped into an FSM shown in Figure 4. The whole process looks like:

Upon receiving an order generation request from Sales, an agent at $state_1$ will generate an order. Coming next is that if this order is accepted, an agent at $state_2$ will submit the order to Accounting. Upon receiving the order, an agent at $state_3$ will check if the payment is on credit. $State_4$ will be reached when an acknowledgment message confirms the credit status. An agent at $state_4$ will approve the credit. Upon receiving the approval of invoice, $state_5$ will be reached which indicates the end of business activities at Accounting. It is also one of the final states in an order fulfillment process. Meanwhile, upon receiving the approval of assembly, the process will move on to $state_6$. After that, the final state, $state_7$ will be reached upon receiving the approval of shipment.

Actually, Figure 4 only describes a simple form of an order fulfillment process. It has disregarded loops which happens in some cases in e-business. For example, at $state_3$, the outcome of this state would be either acknowledgment which leads to $state_4$ as shown in Figure 4, or rejection which may result in a loop between $state_4$ and some previous states or another state which has been omitted from Figure 4.

In addition to the description of process loops in an organisation, an FSM is able to express real processes such as process concurrency or multiple concurrent paths of execution, which are impossible or only with limited support from a traditional modelling approach. In the above case, billing and shipping in Figure 4 are examples. Billing and shipping paths can execute completely independent of one another in the order fulfillment process. Let us take a close look at their executions to illustrate process concurrency. Obviously, any new input string will lead the process into a new state. However, which new state it arrives is dependent on the input signal. On one hand, the billing path is activated when a payment notification from a bank is received. On the other hand, the shipping path will be motivated when a delivery confirmation from a shipper is received,

Apparently, it is essential that participating agents in a process commit to some terminology common to an order fulfillment process. Ontologies are foundations for process automation in e-business. With a most general ontology and other individual ontologies supported by dynamic ontology mapping and integration mechanisms we developed, agents are able to communicate with each other to carry out their defined tasks in an e-business process. FSM agents' reactions to the environment are governed by their embedding processing logic and ontologies at run-time instead of design-time. Furthermore, once an FSM is initiated, FSM agents will operate autonomously. In this sense, agent-based approach is more flexible as it poses as fewer constraints as possible. It also becomes a popular approach for modelling e-services in Internet based business.

4.3 FSM Agents

Work on developing FSM agents is underway. We have developed some agents based on the FIPA-compliant JADE platform (*http://jade.tilab.com/*). All our agents make use of JADE Message Templates in order to perform certain tasks based on certain attributes of the message. For example, an agent can send concepts. When another agent receives a message, it will determine which behaviour to invoke. The way we do this is by using JADE Message Templates. The Message Templates allow us to filter messages based on attributes like `performative type`, `conversation id`, and `sender`, etc.

Agents developed in our prototype possess ontologies and can be applied in process automation. These agents with individual domain-dependent ontologies committing to a general ontology firstly need to reach an agreement about information sharing, then with embedding processing logic, these involved agents are invoked automatically according to the processing logic. Eventually, the order are fulfilled completely with no invention from human users.

5 Conclusion

In this paper we have discussed technical backgrounds such as ontology and software agents for e-business processes in a collaborative environment. As business processes are normally too complex to be designed and implemented by sequential models, the solution is to use a formal approach to cater to highly customer-centric processes modelling, where an execution path is highly dependent on the input and is likely uncertain at design-time. A finite state machine (FSM) is an ideal candidate. Moreover, a business process perfectly maps onto an FSM. Agents, embedding with processing logic, possess ontologies to support knowledge sharing which eventually leads to business automation. Our approach has two advantages. On one hand, it provides a promising way to deal with e-service knowledge sharing by allowing service agents to join and leave the system freely. This is extremely important as far as recent developments in Web services are concerned. On the other hand, modelling complex e-business applications becomes possible with FSM agents and ontologies.

References

[1] Ambroszkiewicz, S., Cetnarowicz, K. and Radko, B., "Enterprise Formation Mechanisms based on Mobile Agents", In Holsten, A. et al. (eds.), *Proc. of the Workshop on Intelligent Agents in Information and Process Management*, KI'98, TZI-Bericht No.9, 1998.

[2] Camarinha-Matos, L. M., "Infrastructures for Virtual Organizations - Where We Are", *Proc. the 9th international conference on Emerging Technologies and Factory Automation*, Lisboa, Portugal, pp. 16-19, 2003.

[3] Cranefield, S., Purvis, M., Mariusz Nowostawski, Peter Hwang, "Ontologies for interaction protocols", *Proc. of the 2nd International Workshop on Ontologies in Agent Systems*, Bologna, Italy, 2002.

[4] Fischer, K., Muller, J. P., Heimig, I. and Scheer, A., "Intelligent Agents in Virtual Enterprises", *Proc. of the 1st International Conference and Exhibition on The Practical Applications of Intelligent Agents and Multi-Agent Technology*, U.K, 1996.

[5] Gruber, T. R., Toward principles for the design of ontologies used for knowledge sharing, KSL-93-04, Knowledge Systems Laboratory, Stanford University, 1993. *http://ksl-web.stanford.edu/.*

[6] Jain, A. K., Aparicio IV, M. and Singh, M. P., "Agents for Process Coherence in Virtual Enterprises", *Communications of the ACM*, 42(3), pp. 62-69, 1999.

[7] Jennings, N., and Wooldridge, M., Agent-oriented software engineering, J. Bradshaw (Ed.), *Handbook of Agent Technology*, AAAI/MIT Press, 2001.

[8] Jenz & Partner: http://www.jenzundpartner.de. Accessed on 3th July 2006.

[9] Laudon, C. K., Laudon, P. J., Management Information Systems: Managing the Digital Firm, 9th ed, Prentice Hall, 2006.

[10] Li, L., Wu B., and Yang, Y., An Ontology-Oriented Approach for Virtual Enterprises, LNCS 3007, Springer-Verlag, pp. 834-843, 2004.

[11] Li, L., Wu, B., and Yang, Y., Refinement for Ontology Evolution in Virtual Enterprises, *Proc. of the 4th International Conference on Electronic Business* (ICEB2004), pp. 696-701, Beijing, China, 2004.

[12] Li, L., Wu, B., and Yang, Y., Semantic Mapping with Multi-Agent Systems, *Proc. of the IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, pp. 54-57, Hong Kong, 2005.

[13] Li, L., Yang, Y., and Wu, B., Agent-Based Approach towards Ontology Refinement in Virtual Enterprises, *Proc. of the 3rd International Conference on Active Media Technology, (AMT 2005)*, pp. 220-225, Japan, 2005.

[14] Li, L., Wu, B., and Yang, Y., Agent-Based Approach for Dynamic Ontology Management, LNCS 3683, Part III, Springer-Verlag, pp. 1-7, 2005.

[15] Li, L., Yang, Y., and Wu, B., Agent-Based Ontology Mapping towards Ontology Interoperability, LNAI 3809, Springer-Verlag, pp. 843-846, 2005.

[16] Oliveira, E. and Rocha, A. P., "Agents' Advanced Features for Negotiation in Electronic Commerce and Virtual Organisation Formation Process", European Perspectives on Agent Mediated Electronic Commerce, Springer-Verlag, 2000.

[17] Papazoglou, P. M., "Agent-oriented Technology in Support of E-Business"',*Communication of the ACM*, 44(4), pp. 71-77, 2001.

[18] P&G. http://www.computerworld.com/printthis /2003/0,4814,77855,00.html. Accessed on 3th July 2006.

[19] Szirbik, N. B., Hammer, D. K., Goossenaerts, J. B. M. and , A. T. M., "Mobile Agent Support for Tracking Products in Virtual Enterprises", AGENT'99, Seattle, USA, 1999.

[20] Tamma, V., Wooldridge, M., Dickinson, I., "An Ontology for Automated Negotiation", AMEC 02 Workshop, 2002.

[21] Wooldridge, M., *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.

[22] Wu, B., Li, L., and Yang, Y., "E-Business Value Process Modelling", *Proc. of the 4th International Conference on Electronic Business* (ICEB2004), pp. 408-413, Beijing, China, 2004.

[23] Wu, B., Dewan, M., Li, L. and Yang, Y., "Supply Chain Protocolling", *IEEE Conference on E-Commerce Technology* (CEC'05), pp. 314-321, München, Germany, 2005.